Attorney's Docket No: 019322-340

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Ingo Molnar	)	
Serial No.:	09/934,738	)	
Filed:	August 22, 2001	)	**Confirmation No.** 9016
Group Art Unit:	2145	)	
Examiner:	Aziziul Q. Choudhury	)	
Attorney Docket:	019322-000340	)	
Title:	EMBEDDED PROTOCOL OBJECTS	)	

Box AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF

Real Party in Interest

Red Hat, Inc. is the real party in interest.

Related Appeals and Interferences

There are no other appeals or interferences, known to the Appellants, or Appellants' legal representatives, which will directly affect or be directly affected by or have a bearing on the Board's decision in this pending appeal.

Status of Claims

Claims 1-14 are pending. The November 1, 2007 final rejection of all pending claims is being appealed herein.

Status of Amendments

There were no amendments filed after the final office action of November 1, 2007. Applicants chose to proceed directly with this appeal. All previous papers filed by Applicants have been entered.

Summary of Claimed Subject Matter

The claimed invention improves the performance of data-serving applications by providing a mechanism to respond to a communication request made by a client application via a dynamic response. The dynamic response is created by mixing together dynamic protocol objects (i.e. the portion of the response that changes) and static protocol objects (i.e. the part of the response that is static) at the server. In effect, the part of the response that is static can be embedded in the dynamic response, along with the portion of the response that changes. If a response, such as a Web page, requires updated content, only a small portion of which has changed, most of the response can be created and sent using the cached static parts, resulting in the response being sent more quickly and efficiently than is possible with prior-art server systems.

The invention allows a communication server to respond to an application protocol request received from a client application so that the response includes at least a portion that changes. The response is designed to be able to also include embedded static parts. In responding to the application protocol request, the portion(s) of the response that changes is sent to the client application. To complete the response, the part(s) of the response that is static is retrieved from cache and sent to the client application. The part(s) of the response that is static is effectively embedded in the dynamic response.

Claims 1, 5, 9, and 11 are commensurate independent claims that stand rejected under the same art. Claim 1 is an independent method claim. The first element of claim 1 is directed to receiving from a client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a portion of the response that is static. This element is shown as 301 in Fig. 3 and discussed in the originally-filed specification in paragraph [0027], page 12, lines 3-10. The second element of claim 1 is directed to creating at the server the portion of the response that changes. This element is shown as 303 in Fig. 3 and discussed in the specification in paragraph [0028], page 12, lines 19-22. The third element of claim 1 is directed to sending the portion of the response that changes to the client application. This element is shown as 304 in Fig. 3 and is also discussed in the specification in paragraph [0028], page 12, lines 22-23. The fourth element of claim 1 is directed to retrieving the portion of the response that is static from a cache disposed in an operating

system kernel. This element is shown as 305 in Fig. 3 and 105 of Fig. 1; also, this element is discussed in the specification in paragraph [0028], page 12, lines 23-25. The fifth element of claim 1 is directed to sending the part of the response that is static to the client application. This element is shown as 306 in Fig. 3 and discussed in the specification in paragraph [0028], page 12, line 25 through page 13, line 2.

Claim 5 is a computer program product claim. It is commensurate with claim 1, discussed immediately above, and thus support and discussion of the operation of the computer program product as claimed occurs in the specification and drawings as discussed above. In addition, portions of the instructions claimed can reside in the system shown in Fig. 6 in processor 602, memory 605, and various media, such as a fixed disk drive 607, a diskette drive 608 and a display 609. This system and the nature of the computer program product, in general, is further discussed in the specification at paragraph [0040], which corresponds to page 18, line 12 through page 19, line 8. The instructions are also shown in the Appendix.

Claim 9 is an apparatus claim directed to responding to a client application. The first element of claim 9 is a cache disposed in an operating system kernel. Support for this element occurs in 105 of Fig. 1 and in paragraph [0022] of the specification (page 7, line 23 through page 8, line 5). The remainder of the elements of claim 9 are recited in means-plus-function language pursuant to 35 U.S.C. § 112 ¶ 6. In each case, the means for performing the function is a combination of a computer system like that shown in Fig. 6 and discussed in paragraphs [0020] and [0025] of the specification and a computer program product as discussed in paragraph [0019] of the specification. The function performed on each element of claim 9 is commensurate to one of the elements of claim 1 and support and discussion for each function can be similarly found.

Claim 11 recites an instruction execution system that is operable as a communication protocol server. The instruction execution system is supported by paragraph [0041] and Fig. 6. The elements of claim 11 are commensurate with those of claim 1, discussed above, and thus support and discussion of the operation of the instruction execution system occurs in the specification and drawings as discussed above.

Grounds of Rejection to be Reviewed on Appeal

Claims 1-14 are rejected as anticipated under 35 U.S.C. § 102(e) as anticipated by U.S. Patent 6,256,712 to Challenger et al. ("Challenger").

Arguments

The Examiner has finally rejected all of Applicants' claims under 35 U.S.C. § 102(e) as anticipated by U.S. Patent 6,256,712 to Challenger et al. Applicants respectfully disagree with the Examiner's characterization of their invention in view of Challenger. It is axiomatic that, in order for a claim to be anticipated, the cited reference must teach every element of the claim, either expressly or inherently. M.P.E.P. § 2131. All of Applicants' claims have recitations for which no corresponding teaching or disclosure can be found in Challenger.

Among others, the following recitations can be found in all of claims 1, 5, 9 and 11 directly, and in all of claims 2-4, 6-8, 10 and 12-14 through dependence. Applicants' claims, in part, recite: 1) "receiving from a client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;" 2) "sending the portion of the response that changes to the client application;" 3) "retrieving the part of the response that is static from a cache disposed in an operating system kernel;" and 4) "sending the part of the response that is static to the client application." Nowhere does Challenger discuss *any* of these claimed limitations. Rather Challenger only discusses maintaining updated caches. *See* Challenger, column 2, lines 53-55.

All of Applicants' claims recite a "request" and a "response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static." Challenger, by contrast, does not even mention a request, let alone discuss responding to requests. Challenger only discusses combining objects into larger objects as an update mechanism for caches. Because Challenger does not disclose, teach or suggest any request or any response to such request, Challenger cannot teach the claimed recitation of "receiving from a client application an application protocol request" nor any claimed recitations that include the response to such request.

All of Applicants' claims also recite that the response includes at least one portion that changes and at least another portion that is static. Challenger does not even mention any static

or changing portions of responses or even Web pages. Challenger only discusses completely static Web pages or completely changing Web pages. Accordingly, Challenger neither discloses "receiving from a client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static." Also, since Applicants' response is a combination of a portion that changes and a part that is static and Challenger doesn't teach such a response, Challenger is neither "sending the portion of the response that changes to the client application" nor "sending the part of the response that is static to the client application."

All of Applicants' claims further recite the retrieval of the part of the response that is static from a cache disposed in an operating system kernel. Applicant is at a loss to find this concept disclosed in Challenger. The portions of Challenger cited by the Examiner discuss either a proxy cache or a processor cache, neither one of which resides in a kernel. A proxy cache resides in user space and a processor cache resides inside the processor hardware. Challenger does not even mention a kernel, let alone an in-kernel cache. The Examiner suggested that a kernel was "inherent" in Challenger because Challenger mentioned a "computer" and all current computers use a kernel. Applicants disagree with this assertion. However, even if this assertion were valid, Challenger still does not mention the very specific concept of retrieving a part of a response that is static from a cache that is disposed in an operating system kernel. For a proper rejection under Section 102, it is not enough for the Examiner to simply analogize specific claim recitations with portions of the cited reference in a conclusory fashion. The Examiner has suggested that some unwritten "spirit of the design" behind a cited reference can be used to reject a claim over the cited reference under Section 102. However, for a proper rejection under Section 102, the Examiner must demonstrate that the **identical invention** is "shown in as **complete detail** as contained in the....claim," and that the elements are "arranged as required by the claim...." M.P.E.P. § 2131. The Examiner has failed to meet this burden.

Additionally, Challenger does not discuss "retrieving the part of the response that is *static*," as recited in the claims. Challenger is only interested in constantly updating data content that has *changed* and validating WebPages on the server. Nowhere does Challenger teach retrieving static content from the memory cache to be sent as a portion of the response.

For at least the above reasons, the Examiner has failed to show that every element of the above-identified claims is present in the art cited. Thus, Applicants' independent claims are patentable over Challenger.

The recitations already discussed in independent claims 1, 5, 9, and 11 are also contained in dependent claims 2-4, 6-8, 10 and 12-14 through their dependence therefrom. Thus, claims 2-4, 6-8, 10 and 12-14 are also patentable for at least the reasons presented above.

Conclusion

For the reasons state above, Applicants respectfully submit that the rejection standing in this application is improper. The Examiner has failed to establish a prima facie case of anticipation under 35 U.S.C. § 102(e). Therefore, Applicants respectfully submit that claims 1-14 are in condition for allowance. Accordingly, the reversal of the rejection of claims 1-14 is respectfully requested.

Respectfully submitted,

Date: 3/11/08

By: R. Brian Drozd

R. Brian Drozd
Registration No. 55,130,
Attorney for Applicants

P.O. Box 13706
Research Triangle Park, NC 27709
Telephone: (919) 286-8000
Facsimile: (919) 286-8199

Appendix

The following is a clean copy of the claims involved in this appeal.

1. In a communication server, a method of responding to a client application, the method comprising the steps of:

receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;

creating at the server the portion of the response that changes;

sending the portion of the response that changes to the client application;

retrieving the part of the response that is static from a cache disposed in an operating system kernel; and

sending the part of the response that is static to the client application.

2. The method of claim 1 wherein the cache disposed within the operating system kernel is a protocol object cache.

3. The method of claim 1 wherein the application protocol request and the reply are formatted according to a hypertext transfer protocol (HTTP).

4. The method of claim 2 wherein the application protocol request and the reply are formatted according to a hypertext transfer protocol (HTTP).

5. A computer program product comprising at least one of a CD-ROM, DVD-ROM, magnetic tape, diskette, magnetic fixed disk and a semiconductor device having computer program code embodied therein, the computer program code for enabling a server to respond to a client application, the computer program code comprising:

instructions for receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;

instructions for creating at the server the portion of the response that changes;

instructions for sending the portion of the response that changes to the client application;

instructions for retrieving the part of the response that is static from a cache disposed in an operating system kernel; and

instructions for sending the part of the response that is static to the client application.

6. The computer program product of claim 5 wherein the cache disposed within the operating system kernel can be a protocol object cache.

7. The computer program product of claim 5 operable to format the application protocol request and the reply according to a hypertext transfer protocol (HTTP).

8. The computer program product of claim 6 operable to format the application protocol request and the reply according to a hypertext transfer protocol (HTTP).

9. Apparatus for responding to a client application, the apparatus comprising:

a cache disposed in an operating system kernel;

means for receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;

means for creating at the server the portion of the response that changes;

means for sending the portion of the response that changes to the client application;

means for retrieving the part of the response that is static from the cache through an operable connection to the cache; and

means for sending the part of the response that is static to the client application.

10. The apparatus of claim 9 wherein the cache can be a protocol object cache.

11. An instruction execution system operable as a communication protocol server, operable to respond to a client application by performing the steps of:

receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a portion of the response that changes and a part of the response that is static;

creating at the server the portion of the response that changes;

sending the portion of the response that changes to the client application;

retrieving the part of the response that is static from a cache disposed in an operating system kernel; and

sending the part of the response that is static to the client application.

12. The instruction execution system of claim 11 further operable as a hypertext transfer protocol (HTTP) server.

13 The instruction execution system of claim 11 wherein the cache can be a protocol object cache.

14. The instruction execution system of claim 12 wherein the cache can be a protocol object cache.